day1:

what is git?

Git is a version control system used to maintain the project.

--Architechture

--installing git

$ sudo apt-get update

$ sudo apt-get install git

to kill any process

sudo kill -9 <PID> (get <PID> from lsof output.

To know the version of installed git

$ git version

to know the location of git

$ which git

configuration settings :

to create a global user name and email

$ git config - - global user.name ravi

$ git config - - global user.email [ravindrakumar3008@gmail.com](mailto:ravindrakumar3008@gmail.com)

$git config - -list

to change from nao editor to vim editor

git config - - global core.editor “vim”

to set the user configarations permantly edit the .gitconfig file

vim .gitconfig

creating a local repository

create/ innitialize a local repository with git init

$git init <reponame/directoryname>

you can see a .git file this the local repo

ex: /home/ravi/myproject/.git/

to convert the local reop into remote repo

$git init bare

day2:

--go to github.com

-- create an account

--create a remote repository

--using git clone

git add

git commit

git push

--git log <filename>

In git a file has 3 states:

i)modified: that means you are modified the file but not commited to local or remote repository

ii)staged : means you added or marked the modified file in its current version which is ready to go in next commit snapshort

(it is a temporary storage area for the modified file )

iii) commited: means that the data is safly stored in our local repository

-- the .git directory is where stores the metadata and object database for your project. This is the most importent part of the git ,and it is what copied when you clone a repository from another computeer.

-- in other VCS when we clone the repository the source code only will get clone

but in Git the source code is cloned including the metadata of the repository.

-- for every commitGIT generate a number called SHA-1 (cheksumming value)vlue , we can use this for forthere use of history.

-- we add the file using

$git add <file name>

now the file is modified and is in stage

-- git reset HEAD <file name >

to unstage the file

evry file in git has two states

i)tracked : tracked files are files that were in last snapshort (in modifid/ ,stage/,commit state)

ii)untracked files : this files the files that were not in last snapshort

--we can have a git remote repository in GIT HUB .com

go and create a github account and create remote repo

Maven:

--Maven is bulid automation tool

--maven buids on plugins concept , it have thousands of plugin

The end to end process of maven at high level

1.developers devlope the code

2.developers /users submit the code into git hub

3.Devops /Build engineer build the code (compiling,packing, .class file ect..) and generates .war file as a final artifacts for java application.

4.once the build is done and the .war files/ binaries are generated it will deploy into the webservers.

5.QA connects to the servers and test the code.

What is maven ?why we use maven?

Maven is a build automatin tool . We use maven to compile the code , build process ,for deployment of the code in to webservers.

Day2 :

--installation

-To run the mavan we install we need JRE (jave run time environment)

(The **Java Runtime Environment** (**JRE**) is a set of software tools for development of Java applications. It combines the Java Virtual Machine (JVM), platform core classes and supporting libraries. **JRE** is part of the Java Development Kit (JDK), but can be downloaded separately.

-the prerequsite to install the maven is install the JDK (**Java Development Kit** (**JDK**) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development.)

1.download and install the JDK

2.download maven and extract

3.set up environment vatialbles

create a file $USER\_HOME/.bashrc

create below environment vairiables

export M2\_HOME= <maven extracted file location>

export JAVA\_HOME= <jdk extracted file location >

export PATH=$M2\_HOME/bin :$JAVA\_HOME/bin :$PATH

ex:

export JAVA\_HOME=/home/ikshvaku/Downloads/jdk1.8.0\_131

export M2\_HOME=/home/ikshvaku/Downloads/apache-maven-3.5.0

export CATALINA\_HOME=/home/ikshvaku/Downloads/apache-tomcat-9.0.0.M21

export STARTUP\_HOME=/home/ikshvaku/Downloads/apache-tomcat-9.0.0.M21

export PATH=$JAVA\_HOME/bin:$M2\_HOME/bin:$CATALINA\_HOME/bin:$STARTUP\_HOME/bin:$PATH

export –userd to swich b/w parent and child process

USER\_HOME (JAVA\_HOME,M2\_HOME ..)– is used as a vairiable to store the home location of the required path

PATH-- used for conviance to set the all path

$ - is used to assain the location to the path

/bin – all commands stored in this folder , so when we run a command linux searches in /bin by default.

To create jar file$ jar -cvf login.jar login.class c- create v- verbose f-file

to extract the jar file

$ jar -xvf login.jar x- extract

to extract the tar file

tar -zxvf <filename.tar.gz>

tar -xvf <filename.tar>

to create tne tarfile

tar -zcvf <filename.tar.gz>

unzip <filename.zip>

to start the tomcat ,first check wheter the tomcat sevice is started or not

ps -ef | grep tomcat

then startup the tomcat service

got to tomcat/bin directory and run

./startup.sh

4.then exit the termial and open

5.verify whether the maven and jdk are install or not

mvn - - version

java - - version

javac - version

day3

build and release end to end process

1.what is compilation

2.packaging java application

3.what is build?

4. buils and development

5.environment

6.DEV ,QA,B&R team ,devops team collabaration and interaction

7.web application development

build means

--detecting the changes automatically

--build the changes i)compileng

ii) packaging

--run test casess

--deploy the packges

--send emails notifications

Maven standerd project layout

flipkart

day4

A Project Object Model or POM is the fundamental unit of work in Maven. It is an XML file that contains information about the project and configuration details used by Maven to build the project. It contains default values for most projects. Examples for this is the build directory, which is target; the source directory, which is src/main/java; the test source directory, which is src/test/java; and so on.

The basic format inside of pom.xml file :

<modelVersion>4.0.0</modelVersion>

<groupId>myproject</groupId>

<artifactId>com.myproject</artifactId>

<version>4.0.0</version>

<packaging>jar</packaging>

<name>com.myproject</name>

<url>http://maven.apache.org</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

building the project

what will hapen when we exicute mvn install commond

--init processs project with initial requisites

ex:setting properities, creating nessesary directories ect..

--validate : validating project’s stucture

--compile : the main and test code

-- run the test casess

--create jar,war files

--copy build artifacts to local repository .M2

note: maven downloads all missing dependencies into .m2 so we need to have internet connection

Dependencies :

if there is any dependencies in application , then the compiler first downloads the dependent appliction and compiles it . After that it compile the main code .

Inheritence : inheritences ueses to use the code multiple times

to generate the pom.xml file

--mnv architype:generate -DartifactID=gamut -DgroupID=com.gamut -DversinID= 4.0.0

to exicute the pom.xml file

--mvn install

End to end build and deployment:

1.ssh [username@Ipaddress](about:blank)

2. clone gamut application source code into build sever from URL: <https://github.com/gamutkar/gamatkat.git>

3. build gamut kart application using

$ mnv install

when we run mvn command by default it will run pom.xml file and runs the given gole (i.e install)

according to maven life cycle, maven exicutes all required build steps Ex: init ,validate,exicute, runtest cases, packaging ,install etc.

mvn install command reads the pom.xml file for build configurations before it exicute its build life cycle build steps

Ex: It checks GAV (groupID ., artifact ID, version), packges (.jar , .ear, ear), downloads all the required dependencies from central or local repositories and compile the source by including dependencies in javac class path.

Based on the package and <final name> atribute it produces the artifact (.jar,.war,.ear) with given name and install to .m2/repository

Assambling its web application and .war is out final artifact, we need to deploy it to some application server like tomcat to make application accessable by end user.

4. Deloy gamut guru.war application to remote server deployment location i.e $tomcat\_home/webapps using below command or automation script

scp gamutkart.war [gamut-ide-nodel@192.168.0.25](mailto:gamut-ide-nodel@192.168.0.25):/home/gamut-ide-node1/apache-tomcat-8.5.11/webaps

4a. After deployment we need to start the server ustin below command

$tomcat\_home/bin/startup.sh

5. tomcat by deafult run on the port no :8080. so we can access the application with url

ex: https:// tomcat server host name or IP: port/application\_root

https:// 192.168.0.25:8080/gamutguru

6.incase any issue inthe application errors will be loged in $/tomcat\_home/logs/catalina.2017-03-24 logfile

we can check this and find out if there are any errors,exceptions we provide this info to developers.

- to connect to the remote server from local server

$ ssh [username@remote](about:blank) server ipaddress

--to copy from one server to another server

$ scp <filename> [username@ipaddress](about:blank) : destination location

--in real time we will have deploy.sh shall script to take care of deployment and backup war/jar files

-- to startup/shoutdown the tomcat server got to /bin directory then run the startup.sh

$ ./startup.sh

Maven features:

1. convention over configuration

2.universal resorce of build practices usrin maven plugins

3.Automatic dependency solution features

4. Maven is not only a build tool, it also project management tool(using maven we can generate a site containg project information ,dependencies, test results , team working and ect..so Maven is also consider as a project management tool )

5.with a minimall less build code we can build any project

6 easy scraching and filetering of artifacts

7.tool portability (eclips ,net besns)

-------------------------------------------------------

jenkins installation

just deploy the jenkins.war file into the webapps folder of apache-tomcat directory

then go to browser then give <http://localhost:8080/jenkins>

then one window will come askin security code

just follow the location show it and copy the code and past .

https://jenkins.io/doc/book/pipeline/

for deployment on remote server

<https://medium.com/@weblab_tech/how-to-publish-artifacts-in-jenkins-f021b17fde71>

https://wiki.jenkins.io/display/JENKINS/Publish+Over+SSH+Plugin

plugins:

publish over ssh plugin

Docker

Docker installation instractions:

(reference link)

https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-16-04

1.curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

2.sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

3.sudo apt-get update

4.sudo docker run -it ubuntu /bin/bash

--apt-get install net-tools

to avoid sudo

5.sudo usermod -aG docker ikshvaku

6.su – ikshvaku

7.id -nG

8.

Q:how to create multiple containers at a time in docker?

-- install cent os in docker contrainer

-- how to create multiple container in docker? Including the configurations? (follow youtube & documents)

to write dockerfile :

https://docs.docker.com/engine/reference/builder/

https://docs.docker.com/engine/userguide/eng-image/dockerfile\_best-practices/

docker build -f /path/to/a/Dockerfile